# Configuring Repositories



Marcel de Vries

@marcelv https://Fluentbytes.com



## Outline



**Managing Git Permissions** 

Design authentication strategies

**Git maintenance** 

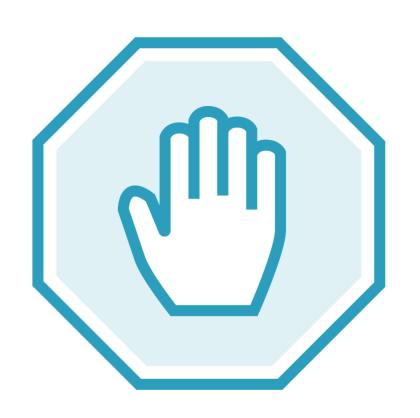
**Summary** 



# Managing Git Permissions



#### Git Permissions



When repository is cloned to your machine, you can access everything

Permissions are set at the server level

- E.g. Branch permissions



# Design Authentication Strategies



## Various Options to Authenticate



Integrating with Azure Active Directory

**Using GitHub SAML integration** 

**Using PAT tokens** 

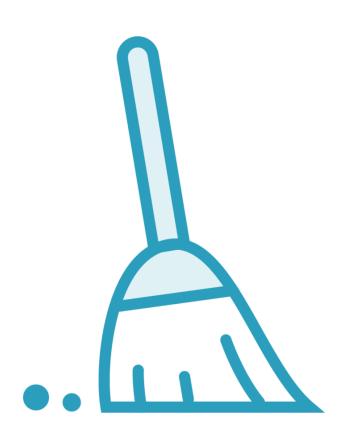
Authorize access to REST APIs with OAuth 2.0



## Git Maintenance



## Maintenance & Recovery



#### Every clone contains the full repository

 git clone downloads the entire history of the project, including every version of every file

# Occasionally, Git automatically runs a command called "auto gc"

- Removes loose objects
- Almost never needed



## Purging a File or Directory



#### More then a new commit!

- Data is always still in the history

#### It requires a rewrite of history

- Commit's hash is changed, along with the hash of all subsequent commits.

#### Use git-filter-branch

- E.g. remove known large file

```
git filter-branch --index-filter "git rm --
cached --ignore-unmatch 'path/file.xyz'"
```



# Demo



Remove a Large File from the Repo



# Summary



Managing Git Permissions

Design authentication strategies

Git maintenance

